AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims

1. (Currently Amended) A code generator for generating an orthogonal code having a spreading factor (SF) and an index (k), wherein the spreading factor (SF) is selectable from values in a range $1 < SF \leq SF_{max}$ with SF_{max} denoting a fixed maximum spreading factor, and wherein the index (k) is in a range $0 < k \leq SF-1$, said code generator comprising:

an index conversion unit for converting the index (k) into a modified index (j) associated with a corresponding code having the fixed maximum spreading factor, wherein the modified index (j) is in a range $0 < j \leq SF_{max}-1$, said index conversion unit comprising:

multiplication means for multiplying the index (k) with a value of SF_{max}/SF, said multiplication means comprising:

a mapping unit for mapping the spreading factor (SF) to a number (s) equal to $\log_2\{SF_{max}/SF\}$; and

a shift register for receiving and storing the index (k) in binary representation, and for receiving the number (s) and shifting the stored index (k) by (s) bit positions in the direction of more significant bit positions; ~~and~~

a permutation unit for permuting the bits of the index (k); and

selection means for selecting, depending upon a mode signal indicating a desired type of said orthogonal code, the output of the permutation unit or the output of the shift register, thereby generating the modified index (j);

a logic unit for solely performing logic operations on bits of the modified index (j) and bits of a counter value (i), thereby generating a code bit of the orthogonal code, wherein the logic unit includes:

adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits; and

combining means for combining the binary output values into the code bit;

a counter for incrementing the counter value (i); and

control means for causing the index conversion unit and the logic unit to sequentially repeat their operations utilizing counter values (i) incremented in steps of one (1) until a desired number of code bits has been generated.

2. (Previously Presented) The code generator according to claim 1, wherein said corresponding code is one of: an orthogonal variable spreading factor (OVSF) code, a Hadamard code, and a Walsh code.

3-7. (Canceled)

8. (Currently Amended) The code generator according to ~~claim 7~~ claim 1, wherein said combining means includes means for performing binary XOR operations.

9. (Canceled)

10. (Currently Amended) A parallel code generator for concurrently generating a number p > 1 orthogonal codes having respective spreading factors (SF_1, ..., SF_p) and indices (k_1,..., k_p), wherein the spreading factors are selectable from values in a range $1 < SF_1, ..., SF_p \leq SF_{max}$, with SF_{max} denoting a fixed maximum spreading factor, said parallel code generator comprising:

a number (p) of code generators, each for generating one of the p orthogonal codes having a particular one of the spreading factors and a particular one of the indices, each of said (p) code generators including:

~~an index conversion unit for converting the index (k) into a modified index (j) associated with a corresponding code having the fixed maximum spreading factor; and~~

~~a logic unit for solely performing logic operations on bits of the modified index (j) and bits of a counter value (i), thereby generating a code bit of the orthogonal code; and~~

~~a counter for generating the counter value (i) to be used by the (p) code generators~~

an index conversion unit for converting the index (k) into a modified index (j) associated with a corresponding code having the fixed maximum spreading factor, wherein the modified index (j) is in a range $0 < j \le SF_{max}-1$, said index conversion unit comprising:

multiplication means for multiplying the index (k) with a value of SF_{max}/SF, said multiplication means comprising:

a mapping unit for mapping the spreading factor (SF) to a number (s) equal to $\log_2\{SF_{max}/SF\}$; and

a shift register for receiving and storing the index (k) in binary representation, and for receiving the number (s) and shifting the stored index (k) by (s) bit positions in the direction of more significant bit positions;

a permutation unit for permuting the bits of the index (k); and

selection means for selecting, depending upon a mode signal indicating a desired type of said orthogonal code, the output of the permutation unit or the output of the shift register, thereby generating the modified index (j);

a logic unit for solely performing logic operations on bits of the modified index (j) and bits of a counter value (i), thereby generating a code bit of the orthogonal code, wherein the logic unit includes:

adding means for performing binary AND operations, wherein the adding means is adapted to receive a bit of the modified index (j) and a bit of the counter value (i), and is further adapted to output a binary output value representing a binary AND combination of the two bits; and

combining means for combining the binary output values into the code bit;

a counter for incrementing the counter value (i); and control means for causing the index conversion unit and the logic unit to sequentially repeat their operations utilizing counter values (i) incremented in steps of one (1) until a desired number of code bits has been generated.

11. (Canceled)

12. (Currently Amended) A method of generating an orthogonal code having a spreading factor (SF) and an index (k), wherein the spreading factor (SF) is selectable from values in a range $1 < SF \leq SF_{max}$, with SF_{max} denoting a fixed maximum spreading factor, and wherein the index (k) is in a range $0 < k \leq SF-1$, said method comprising the steps of:

a) converting by an index conversion unit, the index (k) into a modified index (j) associated with a corresponding code having the fixed maximum spreading factor, wherein the modified index (j) is in a range $0 < j \leq SF_{max}-1$, said converting step including:

a.1) multiplying the index (k) with a value of SF_{max}/SF by mapping the spreading factor (SF) to a number (s) equal to $\log_2\{SF_{max}/SF\}$, and performing a shift register operation comprising receiving and storing the index (k) in binary representation, receiving the number (s), and shifting the stored index (k) by (s) bit positions in the direction of more significant bit positions;

b) initializing a counter value (i);

c) solely performing logic operations by a logic unit on bits of the modified index (j) and bits of the counter value (i), thereby generating a code bit of the orthogonal code, wherein the logic operations include:

c.1) performing binary AND operations, wherein the logic unit receives a bit of the modified index (j) and a bit of the counter value (i), and outputs a binary output value representing a binary AND combination of the two bits; and

c.2) combining the binary output values into the code bit;

d) incrementing the counter value (i) by one; and

e) repeating steps c) and d) until a desired number of code bits has been generated.

a permutation unit for permuting the bits of the index (k); and

selection means for selecting, depending upon a mode signal indicating a desired type of said orthogonal code, the output of the permutation unit or the output of the shift register, thereby generating the modified index (j);

control means for causing the index conversion unit and the logic unit to sequentially repeat their operations utilizing counter values (i) incremented in steps of one (1) until a desired number of code bits has been generated.

13. (Previously Presented) The method according to claim 12, wherein said corresponding code is one of: an orthogonal variable spreading factor (OVSF) code, a Hadamard code, and a Walsh code.

14-16. (Canceled)

17. (Currently Amended) The method according to ~~claim 14~~ claim 12, wherein step a) also includes the steps of:

a.2) permuting the bits of the index (k) by a permutation unit; and

a.3) selecting, depending upon a mode signal indicating a desired type of the orthogonal code, the permuted index or the shifted index, thereby generating the modified index (j).

18. (Canceled)

19. (Currently Amended) The method according to ~~claim 18~~ claim 12, wherein ~~said~~ the combining step c.2) ~~of combining~~ includes performing binary XOR operations.

20. (Canceled)

21. (Previously Presented) A computer-readable medium encoded with a computer program product ~~loaded on an internal memory of a communication unit, comprising software code portions~~ for generating an orthogonal code having a spreading factor (SF) and an index (k), wherein the spreading factor (SF) is selectable from values in a range $1 < SF \leq SF_{max}$, with SF_{max} denoting a fixed maximum spreading factor, and wherein the index (k) is in a range $0 < k \leq SF-1$, wherein the ~~software code portions perform~~ computer program product performs the following steps when ~~the computer program is~~ run on a processor ~~of the communication unit~~:

a) converting by an index conversion unit, the index (k) into a modified index (j) associated with a corresponding code having the fixed maximum spreading factor, wherein the modified index (j) is in a range $0 < j \leq SF_{max}-1$, said converting step including:

a.1) multiplying the index (k) with a value of SF_{max}/SF by mapping the spreading factor (SF) to a number (s) equal to $\log_2\{SF_{max}/SF\}$, and performing a shift register operation comprising receiving and storing the index (k) in binary representation, receiving the number (s), and shifting the stored index (k) by (s) bit positions in the direction of more significant bit positions;

b) initializing a counter value (i);

c) solely performing logic operations by a logic unit on bits of the modified index (j) and bits of the counter value (i), thereby generating a code bit of the orthogonal code, wherein the logic operations include:

c.1) performing binary AND operations, wherein the logic unit receives a bit of the modified index (j) and a bit of the counter value (i), and outputs a binary output value representing a binary AND combination of the two bits; and

c.2) combining the binary output values into the code bit;

d) incrementing the counter value (i) by one; and

e) repeating steps c) and d) until a desired number of code bits has been generated.